SEAMLESS COMPUTER SYSTEM REMOTE CONTROL

BACKGROUND OF THE INVENTION

1. Technical Field:

The present invention generally relates to remotely controlling data processing systems and in particular to employing a service processor within a remotely managed data processing system for remote control. Still more particularly, the present invention relates to employing a service processor to intercept video output and force mouse/keyboard input to a main processor through a connection to a remote console.

2. Description of the Related Art:

With current computer systems, a desire exists to remotely manage one or more systems utilizing standard interconnections--such as a local area network ("LAN") serial connection--and a remote console at which full keyboard and mouse control may be initiated to manipulate the remotely managed systems. Current designs and applications available for this purpose are varied in their abilities and in the areas which they can control. On some systems, the ability to remotely watch and control a managed system during the period of time between when the managed system is powered on until the operating system (OS) begins loading exists only through the use of video and keyboard interrupts for re-direction to a serial port. Additionally, applications such as PC Anywhere or Netfinity Director utilize the Distributed Command Architecture Framework (DCAF) to allow for remote console take-over of managed systems one the operating system is loaded and operational.

Even when combined, however, these two solutions are lacking in several areas. First, in order to remotely manage the system at any time during the system

life, from the time of power on until the operating system is up and running and thereafter, the remote manager is required to initiate one set of protocols or functions during the Post On Self Test (POST) timeframe, then disconnect and re-establish the connection with the remote console once the operating system has loaded. This forces a drop of connection and a separate initiation for extended control.

Second, there remains a window in time--from the point at which the system is powered on until the operating system is up and running--during which the managed system cannot be remotely controlled. As illustrated in the system time line of **Figure 4**, there exists a period of time between the end of the POST Console redirection and the point in time at which the operating system is up and running during which the operating system is loading its kernel and associated device drivers. During this period, when no remote monitoring or control is available under the combination of existing solutions described above, a large number of operating system errors occur. If a critical error occurs during this time period, the remote manager cannot get into the system to view the failing situation because the remote console is switching between remote management applications (i.e., the POST redirection utility has shut down and the remote console application has not yet started).

Third, the current designs require that the remote console include one set of utilities running on the native operating system of the remote console which understand how to communicate with the POST redirection code as well as a second set of utilities which understand how to communicate with the remote console (RC) application. In general, there exists no unified mechanism for a terminal operator to seamlessly--with one application--have full remote control over the managed system from power on until shut down.

Lastly, there exists a requirement that the management applications which currently utilize remote control exists as a specific set of utilities resident on the

remote console. Given that problems may occur with a system at anytime, the ability to remotely control a managed system from anyplace requires the managing individual to have a console with them at all times, with the appropriate software running. In order to allow the managing individual to move from system to system, the remote console should be operable without unique software running on the console.

It would be desirable, therefore, to allow a single connection to be established with a remotely managed system to control that system from power on through operating system load, with a single user interface for remote management throughout this period and without requiring unique management software at the remote console.

SUMMARY OF THE INVENTION

It is therefore one object of the present invention to provide improved remote control over data processing systems.

It is another object of the present invention to employ a service processor within a remotely managed data processing system for remote control.

It is yet another object of the present invention to provide a mechanism for employing a service processor to intercept video output and force mouse/keyboard input to a main processor through a connection to a remote console.

The foregoing objects are achieved as is now described. A remote control application is loaded and executes on a service processor independent from a main processor within a remotely managed system, prior to power on for the main processor. The remote control application grabs and packetizes video data from the remotely managed system for transmission to the remote console via a TCP/IP connection transport layer, and receives keyboard/mouse signals in the same manner for insertion into the remotely managed systems's keyboard/mouse controller(s) as though originating from locally attached peripherals. The service processor also feeds up a Java applet for displaying the video data and capturing keyboard/mouse actions through a browser at the remote console. Remote control is thus enabled from power up of the main processor continuously through operating system load by the main processor with a single user interface, a single connection, and no special utility requirements at the remote console.

The above as well as additional objectives, features, and advantages of the present invention will become apparent in the following detailed written description.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself however, as well as a preferred mode of use, further objects and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

Figure 1 depicts a data processing system network enabling remote control of selected managed systems in accordance with a preferred embodiment of the present invention;

Figure 2 is a block diagram of a remote control system in accordance with a preferred embodiment of the present invention;

Figure 3 is a high level flowchart for a process of remotely controlling a data processing system in accordance with a preferred embodiment of the present invention; and

Figure 4 is a system time line illustrating a combination of existing remote control methods.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

With reference now to the figures, and in particular with reference to **Figure 1**, a data processing system network enabling remote control of selected managed systems in accordance with a preferred embodiment of the present invention is depicted. Data processing system network **102** includes a remote console system **104** coupled via a network **106** to one or more remotely managed systems **108a-108n**. Remote console system **104** need not be different in construction and operation from remotely managed systems **108a-108n**, but may instead simply be one of a number of interconnected systems, all subject to remote control, which is currently executing a remote console application.

The structure and operation of data processing systems **104** and **108a-108n**, as well as of network **106**, is well-known in the art, and only so much of that construction and operation which is unique to the present invention and/or necessary for an understanding of the present invention is described herein.

Referring to **Figure 2**, a block diagram of a remote control system in accordance with a preferred embodiment of the present invention is illustrated. The present invention solves the problem of remote control by taking advantage of a service processor **202**, independent of the main processor (not shown), within the remotely managed system **108n**. In contemporary designs, service processors are being incorporated into various types of data processing systems, from high-end servers to set top boxes, for a variety of other purposes. Given a system with an independent service processor **202**, the firmware of the service processor **202** is capable of gaining control of the managed hardware and acting as a conduit for the remote control application to a management console. In the present invention, service processor **202** serves four distinct functions:

First, the service processor **202** has control over the keyboard/mouse controller(s) **204** within the remotely managed system **108n**. The service processor **202** has the ability to receive keyboard and mouse data remotely, and to force that data into the system keyboard/mouse controller(s) **204** to create the appearance, to the remotely managed system **108n**, that the remotely managed system **108n** received real keyboard and mouse data from locally attached peripherals.

Second, the service processor **202** must have the ability to get video data out of the video hardware **206** within the remotely managed system **108n**. This function may be performed in several known ways, including video snooping of the actual video hardware or video redirection via device drivers or firmware.

Third, the service processor **202** should be able to communicated with a remote console **104** via an industry standard communication packet. In the preferred embodiment, Transmission Control Protocol/Internet Protocol (TCP/IP) is employed to allow a generic Web browser to be employed at the remote console **104** in controlling remotely managed system **108n**. As a result, no unique set of software executing on the remote console **104** is required to control remotely managed system **108n**. Network **106** thus includes operation through a TCP/IP stack **208** and a Point-to-Point Protocol (PPP) stack **210**.

Lastly, the service processor **202** should, in the preferred embodiment, be capable of serving up a Java applet to the Web browser running within the remote console system **104**. The Java applet should receive video data from the remotely managed system **108n** and display that video data on the screen at the remote console **104**, and capture keyboard and mouse inputs from the remote console system **104** for redirection to the remotely managed system **108n**. The Java application may employ push technology for the video from the remotely managed system **108n** up to the

remote console **104** as well as pull technology of the keyboard and mouse inputs from the remote console **104** down to the remotely managed system **108n**.

Given the environment depicted and described above, the service processor **202** is able to accommodate the requirements for remote control from power on through operating system load as outlined above. A remote control power on request is initiated from the service processor **202**, when, for example, power is turned on at the remotely managed system **108n**. The service processor **202** first loads the remote control application which allows the service processor **202** to receive and manage remote control information (e.g., intercept video data and direct keyboard/mouse signals), then serves up the remote console Java applet to the Web browser within the remote console **104**. The service processor **202** will then reset or power on the remainder of the remotely managed system **108n**.

Once POST is started during power on of the main processor within the remotely managed system **108n**, the remote control application executing in the service processor **202** will begin grabbing the video information from the host, packetizing that video information, and transmitting the packetized video information through the TCP/IP transport layers to the remote console **104**. The java applet running in the Web browser on the remote console **104** will remove the video data from the packet(s) and build video screens on the display terminal for the remote console **104**. The reverse course is concurrently maintained for keyboard/mouse control, with the remote console **104** packetizing the keyboard and mouse data which is sent back to the service processor **104** over the TCP/IP link, where the service processor **104** stuffs the keyboard and mouse data back into the hardware controller(s) within the remotely managed system **108n**.

With reference now to **Figure 3**, a high level flowchart for a process of remotely controlling a data processing system in accordance with a preferred

embodiment of the present invention is depicted. The process begins at step **300**, which depicts the service processor within a remotely managed system receiving a remote control power on request, originating either with the remotely managed system being turned on or from a remote console desiring to initiate remote control. The process first passes to step **302**, which illustrates loading the remote control application for execution by the service processor, and then to step **304**, which illustrates the service processor serving a Java applet for the remote control user interface to the remote console utilizing a TCP/IP connection.

The process next passes to step **306**, which depicts the service processor powering on or resetting the main processor within the remotely managed system. The process then passes to step **308**, which illustrates grabbing the video data for the remotely managed system and packetizing that video data for transmission over the TCP/IP connection, all utilizing the service processor. It should be noted that the video may be acquired at regular intervals any time during the period following initiation of the POST routine, including the time during which the operating system kernel and drivers are being loaded. The process passes next to step **310**, which depicts receiving packetized keyboard/mouse signals over the TCP/IP connection and, utilizing the service processor, forcing the received keyboard/mouse signals into the keyboard/mouse hardware controller within the remotely managed system. Again, it should be noted that the keyboard/mouse signals may be redirected through the local controller(s) any time during the period following initiation of the POST routine, including the time during which the operating system kernel and drivers are being loaded.

The Java applet served by the service processor to the remote console provide a user interface displaying the video data from the remotely managed system, and intercepts keyboard and mouse actions for packetizing and transmission to the service processor. In effect, the Java applet duplicates the video terminal and the

keyboard/mouse ports at the remote console.

The process next passes to step **312**, which illustrates a determination of whether remote control of the remotely managed system has been terminated. If not, the processor returns to and repeats steps **308** and **310**, continually sending video data from the remotely managed system to the remote console and inserting keyboard/mouse signals received from the remote console into the keyboard/mouse controller(s) for the remotely managed system. Once remote control is terminated, the process passes to step **314**, which illustrates the process becoming idle until remote control is again initiated for the remotely managed system.

The present invention provides a remote control capability which is independent of the operating system within the remotely managed system and allows for total remote control, from power up of the remotely managed system through loading of the operating system within the remotely managed system, and while that operating system is running. Additionally, the present invention provides a mechanism which allows the service processor to feed up the requisite remote management application to a standard Web browser utilizing standard communications protocols, allowing the remote console operator to be located essentially anywhere with any system running a standard Web browser and a connection to the remotely managed system, without requiring a special remote control application or utility.

The present invention solves the problems left by the combination of existing remote control solutions described above, and allows for a single connection to be established with the remotely managed system in order to control the system from power on through operating system load, and while the operating system is running. The present invention also provides a single user interface for remote management through the time cycle of the remotely managed system from power on through

operating system load and beyond, and enables the requisite management application to be served up to the remote console from the remotely managed system, not requiring a unique piece of management software at the remote console.

It is important to note that while the present invention has been described in the context of a fully functional data processing system and/or network, those skilled in the art will appreciate that the mechanism of the present invention is capable of being distributed in the form of a computer usable medium of instructions in a variety of forms, and that the present invention applies equally regardless of the particular type of signal bearing medium used to actually carry out the distribution. Examples of computer usable mediums include: nonvolatile, hard-coded type mediums such as read only memories (ROMs) or erasable, electrically programmable read only memories (EEPROMs), recordable type mediums such as floppy disks, hard disk drives and CD-ROMs, and transmission type mediums such as digital and analog communication links.

While the invention has been particularly shown and described with reference to a preferred embodiment, it will be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention.